A method of and apparatus for monitoring event logs

Field of the invention

[0001] The present invention relates generally to the field of data processing, and more particularly without limitation, to event log monitoring.

Background and prior art

[0002] The process of recording events is referred to as "event logging", a terminology adopted from the meticulous practice that a ship's captain uses to enter daily notes during a sea voyage. In the electronic world, events are logged in storage devices and later used to derive some desired information concerning usage and operation of the system.

[0003] Some computer operating systems have an event logging component. The Windows operating system from Microsoft Corporation logs events which reflect operation of the computer system. The events are logged locally to a storage, such as the hard disk drive, that is resident on the same computer that the operating system is running.

[0004] Typically event logs are checked by the system administrator after a problem or malfunction occurred in order to identify the cause of the problem. Such a manual checking procedure is a tedious task. Therefore various methods for automatic monitoring of event logs have been devised in the prior art:

[0005] US patent no. 5,867,659 shows an event log forwarder which accesses a set of one or more filters and checks whether a new event in one or more event logs satisfies the set of one or more filters. The event log forwarder also provides an indication if there is a new event which satisfies the set of one or more filters. Addition-

ally, the event log forwarder automatically repeats, at periodic intervals, checking whether a new event in one or more event logs satisfies the set of one or more filters and provides an indication if there is a new event which satisfies the set of one or filters.

[0006] US patent no. 6,347,335 shows a common event log for a distributed computer system including a plurality of computer nodes. The common event log includes a plurality of storage locations for storing common event log entries. Each computer node performs processing operations in connection with a program, and generates, at selected points in its program, an event log entry including status information representing status of the computer node at the point at which the log entry was generated, the computer nodes storing the event log entries which they generate in the common event log contemporaneous with the generation thereof. As a result, the event log entries are stored in the common event log in the order in which the computer nodes reach the points in their respective programs. The common event log includes a buffer comprising a plurality of storage locations, and the location at which an entry is to be stored is pointed to by a write pointer.

[0007] US patent no. 6,507,852 shows an location-independent service for monitoring and alerting on an event log. For monitoring of the event log one or more alert policies are accessed, wherein each of the alert policies is comprised of one or more rules stored on a computer. An event log stored on a computer is accessed in a location-independent manner to gather one or more event messages stored therein. The event messages are filtered by comparing them to the rules of the alert policies to raise an alert and determine whether an alert action should be invoked.

Summary of the invention

[0008] The present invention provides for a method of monitoring a plurality of local event logs of a computer network. The local event logs are entered into a central database of the computer network. The central database is sent from the computer network to an external support computer system for analysis of the local event logs.

[0009] In accordance with a preferred embodiment of the invention the node identifiers of the network nodes are used as keys for storing of the local event logs in the central database. This enables the external support computer system to analyse the individual local event logs stored in the central database with respect to individual ones of the network nodes.

[0010] In accordance with a further preferred embodiment of the invention the central database resides on a server computer of the computer network. The local event logs are transmitted from the network nodes to the server computer and are stored in the central database. Preferably the server computer has a local server event log which is also stored in the central database.

[0011] In accordance with a further preferred embodiment of the invention the transmission of the local event logs from the network nodes to the server computer is initiated by the server computer. This can be done by remote execution of program code which is provided from the server computer to the network nodes.

[0012] In accordance with a further preferred embodiment of the invention a discovery procedure is carried out prior to transmission of the local event logs to the server computer. In the discovery procedure the network

topology, network node configurations and / or other data is determined by the server computer. The network topology information and configuration information can be utilized by the server computer to collect the local event logs from the network nodes.

[0013] In accordance with a further preferred embodiment of the invention the central database is sent from the server computer of the customer computer network to the external support computer system at periodic time intervals which are customisable. The external support computer system performs an analysis of the local event logs stored in the central database and generates an alert message if a potential problem is identified. Preferably the analysis is performed by means of a rule base of alert policies.

[0014] In accordance with a further preferred embodiment of the invention the external support computer system performs as database query in order to identify the last "send event" which has been entered into the local server event log. The "send event" indicates when a previous transfer of the central database to the external support computer system occurred.

[0015] The time stamp of the "send event" is used by the external support computer system to perform another database query in order to identify those local event log entries having time stamps after the "send event" time stamp. In other words the external support computer system determines those local event log entries which are new, i.e. which have not been included in a central database which has been received previously. This way it is prevented that alert messages are generated for past events which had already been analysed in a previous event log analysis.

[0016] In accordance with a further preferred embodiment of the invention the external support computer system generates an alert message for a response center engineer and sends the alert message as an email to an email address of the response center engineer if an alert condition is detected.

[0017] In accordance with a further preferred embodiment of the invention the external support computer system is used as a response center for servicing a plurality of customer computer networks. The response center computer receives central databases containing local event logs from the various customer computer networks for event log analysis.

Brief description of the drawings

[0018] In the following preferred embodiments of the invention will be described, by way of example, and with reference to the drawings in which:

[0019] Figures 1a and 1b, together, is a block diagram of a computer network having a server computer for storing of local event logs in a central database,

[0020] Figure 2 is a block diagram of a support computer system for analysis of local event logs stored in the central database,

[0021] Figures 3a and 3b, together, is illustrative of a flowchart of a preferred embodiment of a method of the invention,

[0022] Figure 4 is illustrative of local event logs stored in a central database.

Detailed description

[0023] Figures 1a and 1b show a computer network 100. Computer network 100 has various network nodes including client computers 102, 104, ... and server computer 106. For example computer network 100 is a local area network (LAN).

[0024] Client computer 102 has central processing unit (CPU) 108 and memory 110. For example client computer 102 uses a Windows operating system which generates local event log 112; local event log 112 is stored locally on client computer 102. Events like starting, finishing or manually stopping an application program or execution of other actions are stored in local event log 112. Each entry into local event log 112 has a text string being descriptive of an event and an event identification number. Further each entry in local event log 112 is time stamped when it is entered in local event log 112.

[0025] In the example considered here an event has been entered into local event log 112 when the Norton AntiVirus application program has been started. Event identification number 01 is assigned to this event and a corresponding entry is made into local event log 112 by the operating system. This entry is time stamped with time T_1 on which the event occurred.

[0026] Likewise an entry into local event log 112 is made when the Frontbase Database program started at time T_2. Subsequently a number of other events is entered into local event log 112.

[0027] Depending on the customizing settings of the Windows operating system past events which are likely of not being of interest to the network administrator anymore are automatically erased from the local event log 112 in

order to limit the size of local event log 112. This can be done by using a predefined time window to remove old event log entries.

[0028] The other client computers 104, ... of network 100 have a similar design.

[0029] Server computer 106 has CPU 114 and memory 116. Further server computer 106 has control program 118, remote execution program 120 and discovery program 122.

[0030] Control program 118 can start discovery program 122 in order to initiate a discovery procedure for the network nodes of network 100 and it can initiate the transfer of the local event logs 112 from the client computers 102, 104,... to the server computer 106 for storage in central database 124.

[0031] Preferably server computer 106 also runs a Windows operating system which creates local server event log 126.

[0032] Server computer 106 has interface 128 for sending of central database 124 to support computer system 130 over network 132. Support computer system 130 has a corresponding interface 134 for receiving of central database 124 from server computer 106 over network 132. For example network 132 is the Internet and the interfaces 128 and 134 are adapted for communication over the Internet.

[0033] In operation an entry is created in local server event log 126 each time a transfer of central database 124 to support computer system 130 occurs. The corresponding entry is made into local server event log 126 after central database 124 has been sent out from server computer 106. In the example considered here a previous transfer of central database 124 occurred at time T_T which

was entered as event entry # 02 in local server event log 126.

[0034] Control program 118 periodically starts discovery program 122 for discovery of the network nodes of computer network 100, including client computers 102, 104,... After completion of the discovery procedure control program 118 initiates the transmission of the local event logs 112 from the client computers 102, 104, ... to server computer 106 over network 100 by transmitting of remote execution program 120 to clients 102, 104, ...

[0035] When remote execution program 120 is remotely executed on clients 102, 104, ... by server computer 106 the event logs 112 stored on client computers 102, 104, ... are transmitted over network 100 to server computer 106 and stored in central database 124. The respective node IDs of client computers 102, 104, ... are used as keys for storing of the respective event log entries. Further, local server event log 126 is also stored in central database 124.

[0036] Next control program 118 sends central database 124 to support computer system 130 over network 132. After completion of this "send event" a corresponding entry is made in local server event log 126 with a time stamp indicating when central database 124 was sent out. This procedure is repeated at customisable periodic time intervals.

[0037] Figure 2 shows a more detailed block diagram of support computer system 130. Support computer system 130 has storage 136 for storing central databases of the type of central database 124 as shown in figure 1. Typically support computer system 130 provides network support services for a plurality of customers i, j, ...Storage 136 has sufficient capacity for storing of a plurality of central databases 124 received from the various customer com-

puter networks of the type of computer network 100 as depicted in figure 1.

[0038] Further support computer system 130 has database query program 138, event log analysis program 140 for performing an analysis of the event logs stored in one of central databases 124 in accordance with rules stored in rule base 142, automatic notification program 144 for sending out a message to a response center engineer in case an alert situation is detected, and memory 146 for storing of data sets to be analysed by event log analysis program 140.

[0039] In operation support computer system 130 receives a sequence of central databases 124 from various customers i, j, ... These central databases 124 are stored in storage 136. Preferably the central databases 124 are processed sequentially in the order of arrival; alternatively the central databases 124 are processed in parallel if processing unit (PU) 148 of computer system 130 has parallel processing capabilities.

[0040] For processing of central database 124 received from server computer 106 (cf. figure 1) of customer i database query program 138 is started in order to retrieve a "send entry" from central database 124 with the latest time stamp. This time stamp indicates the point of time when a previous sent action of central database 124 had been performed by server computer 106.

[0041] Next database query program 138 queries central database 124 received from customer i in order to identify those data sets having a time stamp later than the previous "send entry" time. These data sets are stored in memory 146 for analysis by event log analysis program 140.

[0042] The advantage of determining the previous "send entry" time is that this way those data sets which have been entered after the previous send action are identified. This prevents that the same data sets are analysed each time a new copy of central database 124 is received from customer i.

[0043] The data sets which are stored in memory 146 are analysed by event log analysis program 140 in accordance with rules stored in rule base 142. These rules reflect corresponding alert policies for identification of a potential problem of computer network 100 (cf. figure 1) of customer i. If such a potential problem is detected automatic notification program 144 is invoked in order to send a corresponding message to a response center engineer.

[0044] Figures 3a and 3b, together, show a corresponding flowchart. In step 300 local event logs are received by a server computer of a customer computer network. The local event logs which are received from the network nodes are stored in a database using the node identifiers (ID) of the network nodes as respective keys. This is done in step 302.

[0045] In step 304 the local event log of the server computer is also stored in the database using the node ID of the server computer as a key. Next the database is sent from the server computer to an external support computer in step 306. Preferably steps 300 to 306 are initiated by the server computer at customisable periodic intervals.

[0046] In step 308 the database is received by the external support computer. In step 310 a database query is performed by the support computer in order to identify a "send event" log entry which was entered for a send event of the database from the server computer to the external

support computer prior to the transfer of step 306. The corresponding "send event" time stamp of the send event log entry is used in step 312 in order to carry out a database query for determination of all event log entries stored in the database which have a time stamp which is later than the "send event" time stamp. This way a differential set of event log entries is created. The differential set of event log entries comprises all event log entries which have been added to the central database 124 after the previous database transfer.

[0047] In step 314 the event log entries comprised in the differential set are analysed by means of rules which define a set of alert policies. This way potential problems are identified. If such a potential problem is identified an automatic notification is sent to an administrator or response center engineer. Preferably a corresponding email message containing a description of the identified potential problem and / or of the corresponding event log entries is generated and sent automatically to the response center engineer. The response center engineer can then contact the corresponding customer to which the identified potential problem relates for corrective action.

[0048] Figure 4 shows a set 400 of event log entries of a network node XY. When the Norton AntiVirus program was started on network node XY a corresponding event log entry is generated and stored in the local event log of node XY. The event log ID is 57; when the event log ID was created it was time stamped at time T_{57}.

[0049] Further set 400 which is stored in central database 124 contains an event being descriptive of the termination of the Norton AntiVirus program by either finishing or manually stopping the Norton AntiVirus application program. The corresponding event is entered with

event identifier 63 and time stamp T_{63}. Further set 400 contains other event log entries relating to other application programs. From set 400 it appears that with respect to the Norton AntiVirus application program no problem occurred as the Norton AntiVirus application program was normally started and terminated.

[0050] Set 402 stored in central database 124 contains a set of event log entries being related to network node XZ. Event with event identifier 36 was entered when the Frontbase Database program was started at time T_{36}. Event number 48 indicates that Frontbase Database was started again at time T_{48}. Between events 36 and 48 Frontbase Database was not terminated. This indicates that an abnormal situation may be present and an alert message is generated by the system.

list of reference numerals

100	computer network
102	client computer
104	client computer
106	server computer
108	central processing unit
110	memory
112	local event log
114	central processing unit
116	memory
118	central program
120	remote execution program
122	discovery program
124	central database
126	local server event log
128	interface
130	support computer system
132	network
134	interface
136	storage
138	database query program
140	event log analysis program
142	rule base
144	automatic notification program
146	memory
148	processing unit
400	set
402	set